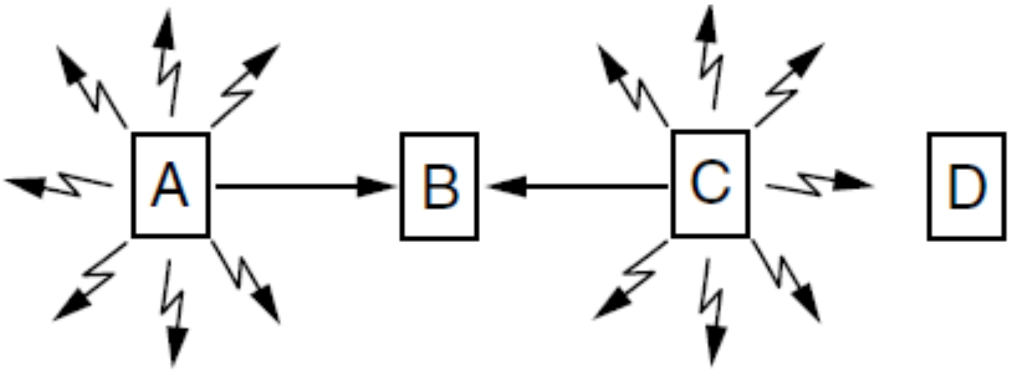
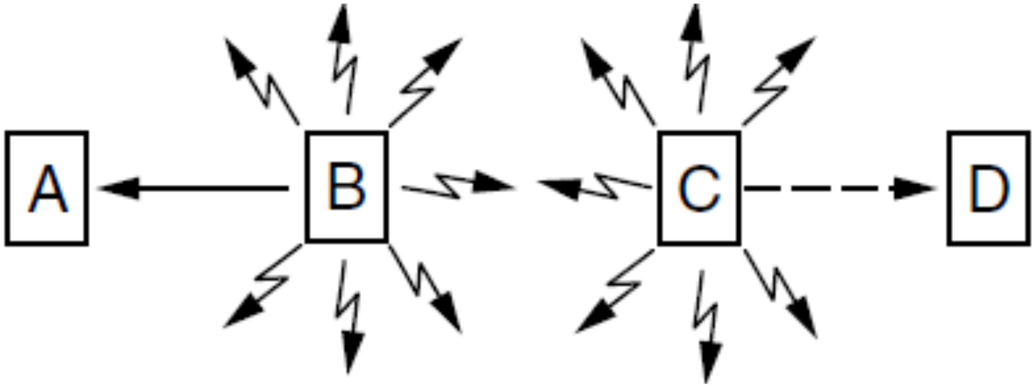
Hidden Terminals

- Nodes A and C are <u>hidden terminals</u> when sending to B
  - Can't hear each other (to coordinate) yet collide at B
  - We want to avoid the inefficiency of collisions



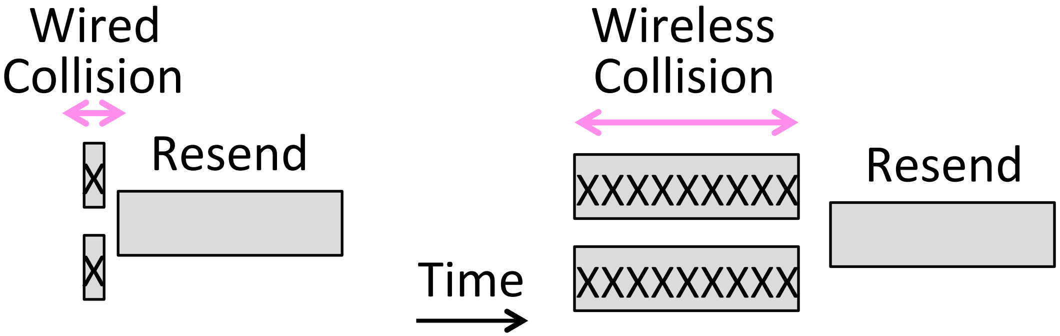
Exposed Terminals

- B and C are <u>exposed terminals</u> when sending to A and D
  - Can hear each other yet don't collide at receivers A and D
  - We want to send concurrently to increase performance



Nodes Can’t Hear While Sending

- With wires, detecting collisions (and aborting) lowers their cost
- More wasted time with wireless



Possible Solution: MACA

- MACA uses a short handshake instead of CSMA (Karn, 1990)
  – 802.11 uses a refinement of MACA (later)

- Protocol rules:
  1. A sender node transmits a RTS (Request-To-Send, with frame length)
  2. The receiver replies with a CTS (Clear-To-Send, with frame length)
  3. Sender transmits the frame while nodes hearing the CTS stay silent
  – Collisions on the RTS/CTS are still possible, but less likely

MACA – Hidden Terminals

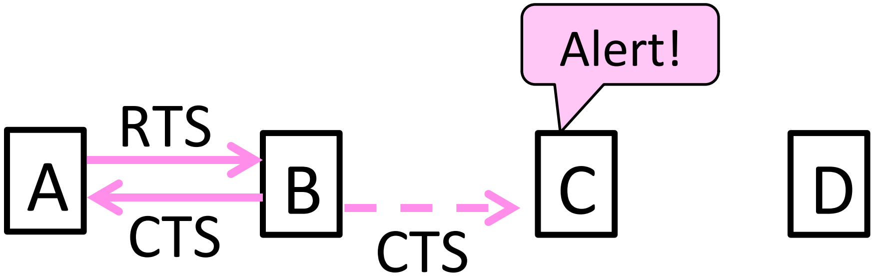
- A→B with hidden terminal C
  1. A sends RTS, to B

MACA – Hidden Terminals (2)

- A→B with hidden terminal C
  2. B sends CTS, to A, and C too

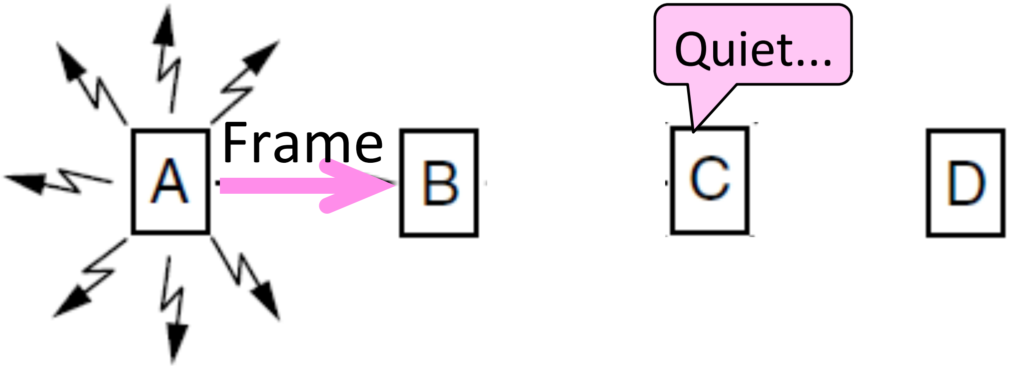
MACA – Hidden Terminals (3)

- A→B with hidden terminal C
  2. B sends CTS, to A, and C too



MACA – Hidden Terminals (4)

- A→B with hidden terminal C
  3. A sends frame while C defers

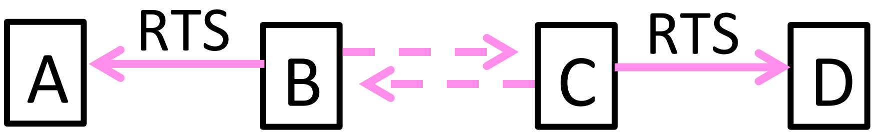


MACA – Exposed Terminals

- B→A, C→D as exposed terminals
  - B and C send RTS to A and D

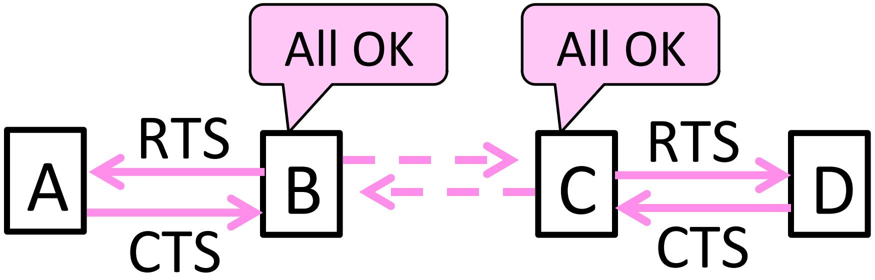
MACA – Exposed Terminals (2)

- B→A, C→D as exposed terminals
  - A and D send CTS to B and C



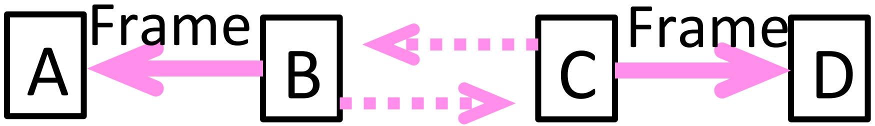
MACA – Exposed Terminals (3)

- B→A, C→D as exposed terminals
  - A and D send CTS to B and C



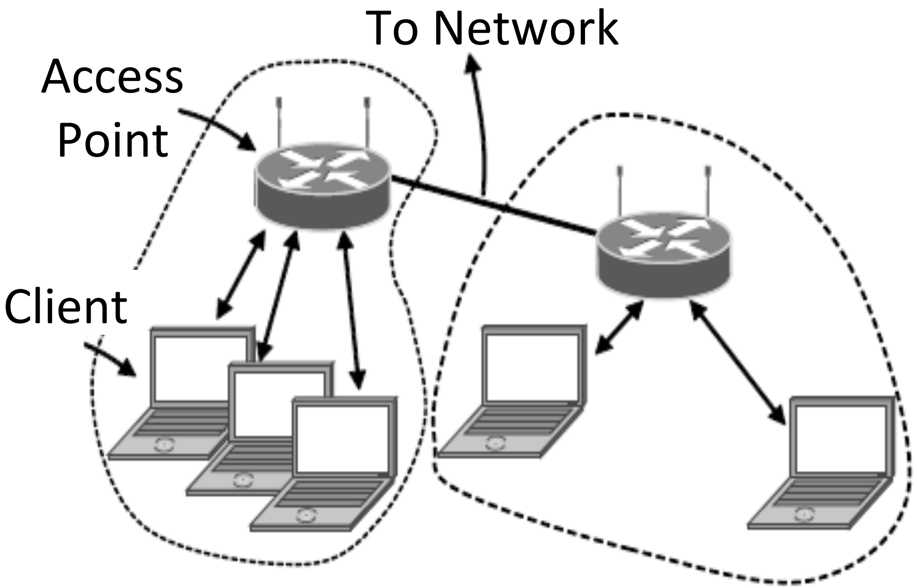
MACA – Exposed Terminals (4)

- B→A, C→D as exposed terminals
  - A and D send CTS to B and C



802.11, or WiFi

- Very popular wireless LAN started in the 1990s
- Clients get connectivity from a (wired) AP (Access Point)
- It's a multi-access problem ☺
- Various flavors have been developed over time
  - Faster, more features

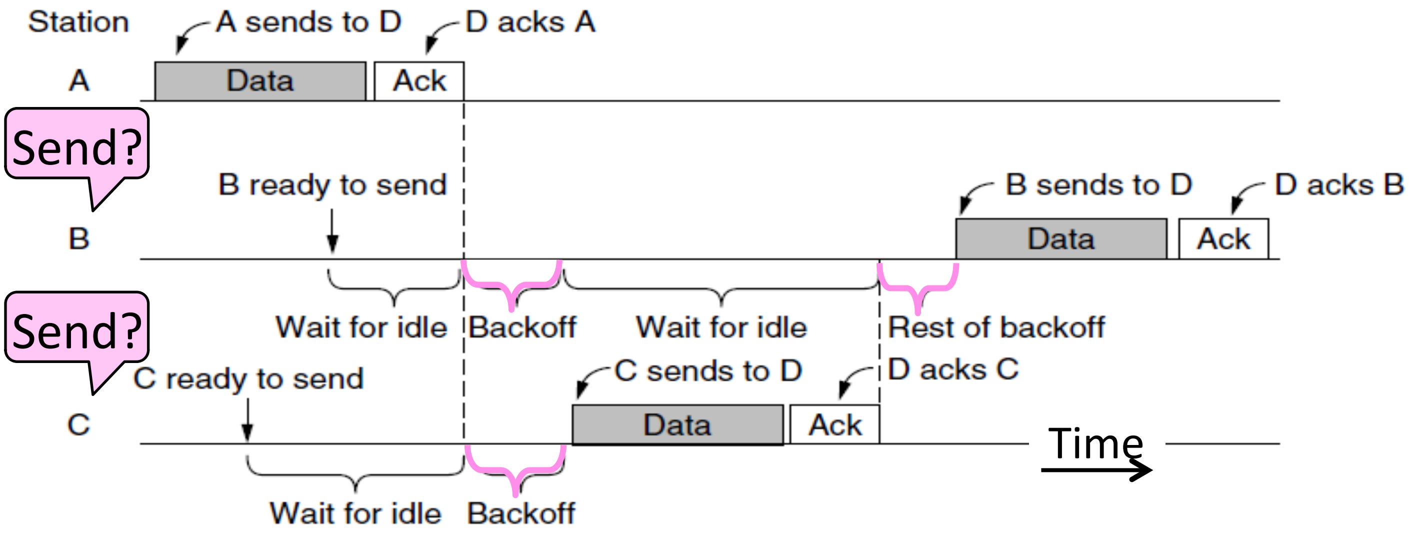


802.11 Physical Layer

- Uses 20/40 MHz channels on ISM bands
  - 802.11b/g/n on 2.4 GHz
  - 802.11 a/n on 5 GHz
- OFDM modulation (except legacy 802.11b)
  - Different amplitudes/phases for varying SNRs
  - Rates from 6 to 54 Mbps plus error correction
  - 802.11n uses multiple antennas; see “802.11 with Multiple Antennas for Dummies”

802.11 CSMA/CA for Multiple Access

- Sender avoids collisions by inserting small random gaps
  - E.g., when both B and C send, C picks a smaller gap, goes first



The Future of 802.11 (Guess)

- Likely ubiquitous for Internet connectivity
  - Greater diversity, from low- to high-end devices
- Innovation in physical layer drives speed
  - And power-efficient operation too
- More seamless integration of connectivity
  - Too manual now, and limited (e.g., device-to-device)

Issues with Random Multiple Access

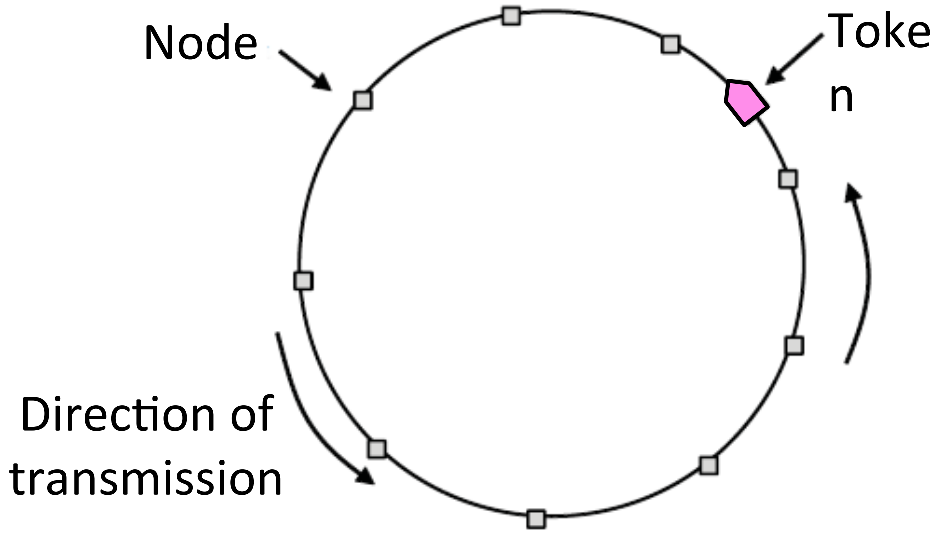
- CSMA is good under low load:
  - Grants immediate access
  - Little overhead (few collisions)
- But not so good under high load:
  - High overhead (expect collisions)
  - Access time varies (lucky/unlucky)
- We want to do better under load!

Turn-Taking Multiple Access Protocols

- They define an order in which nodes get a chance to send
  - Or pass, if no traffic at present
- We just need some ordering …
  - E.g., Token Ring »
  - E.g., node addresses

Token Ring

- Arrange nodes in a ring; token rotates “permission to send” to each node in turn



Turn-Taking Advantages

- Fixed overhead with no collisions
  - More efficient under load
- Regular chance to send with no unlucky nodes
  - Predictable service, easily extended to guaranteed quality of service

Turn-Taking Disadvantages

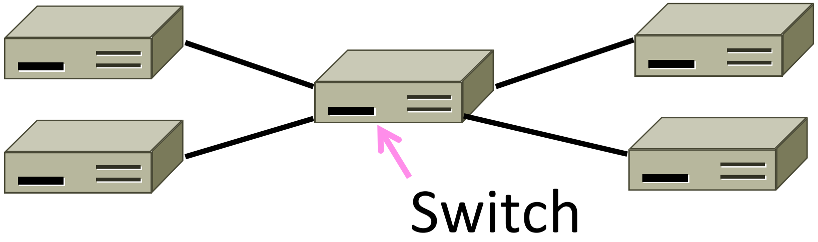
- Complexity
  - More things that can go wrong than random access protocols!
    - E.g., what if the token is lost?
  - Higher overhead at low load

Turn-Taking in Practice

- Regularly tried as an improvement offering better service
  - E.g., qualities of service
- But random multiple access is hard to beat
  - Simple, and usually good enough
  - Scales from few to many nodes

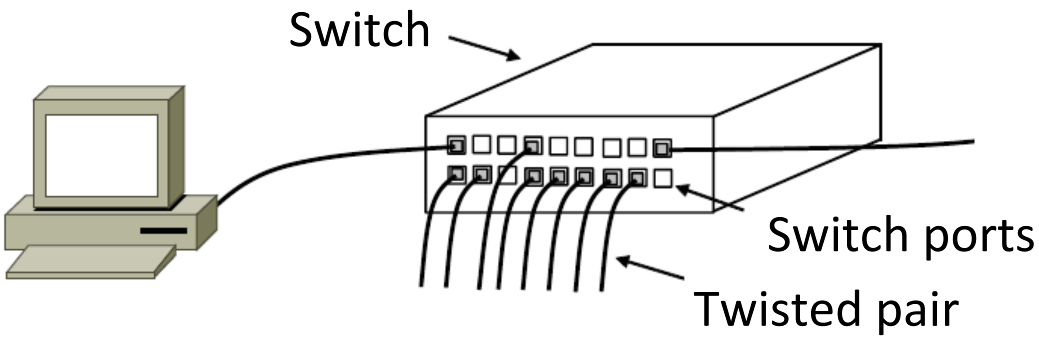
Topic

- How do we connect nodes with a switch instead of multiple access
  - Uses multiple links/wires
  - Basis of modern (switched) Ethernet



Switched Ethernet

- Hosts are wired to Ethernet switches with twisted pair
  - Switch serves to connect the hosts
  - Wires usually run to a closet



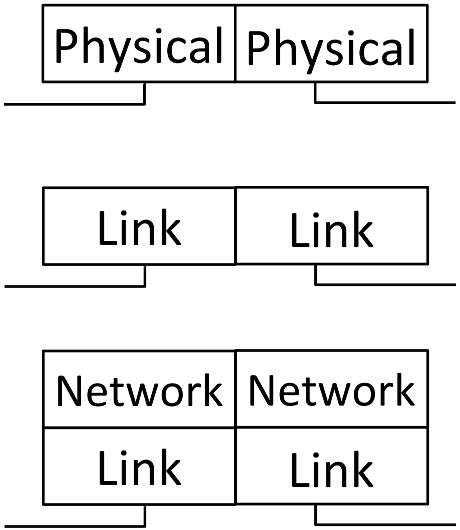
What’s in the box?

- Remember from protocol layers:

Hub, or repeater

Switch

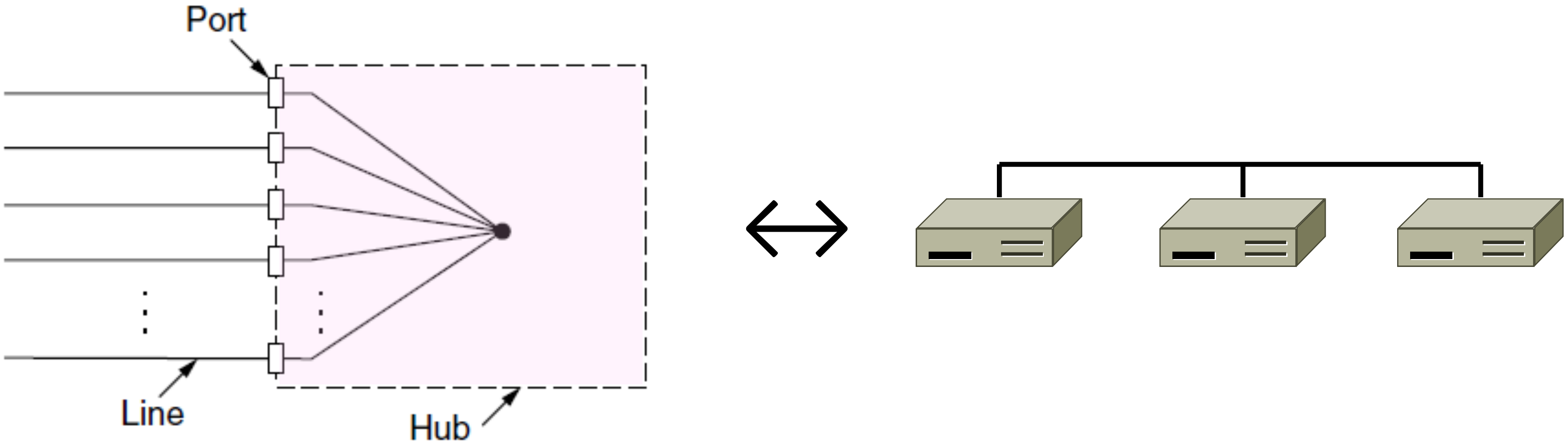
Router



All look like this:

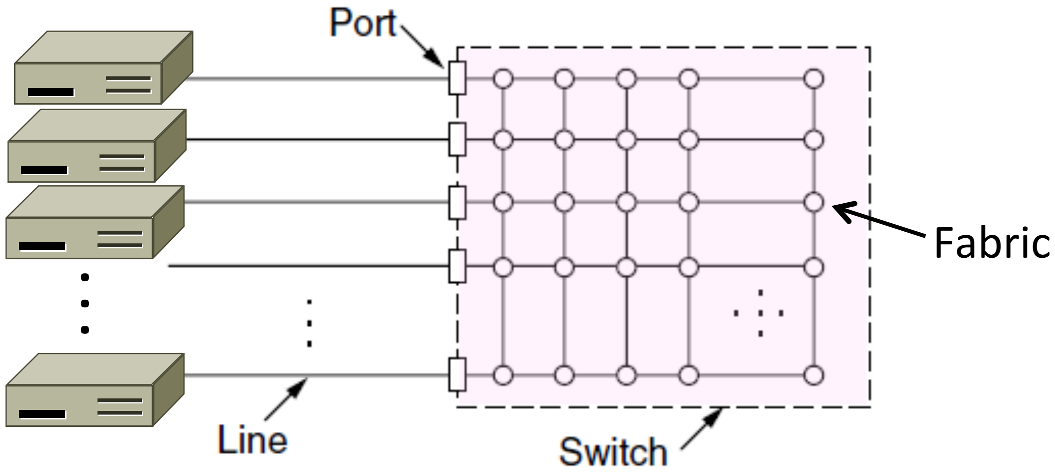
Inside a Hub

- All ports are wired together; more convenient and reliable than a single shared wire



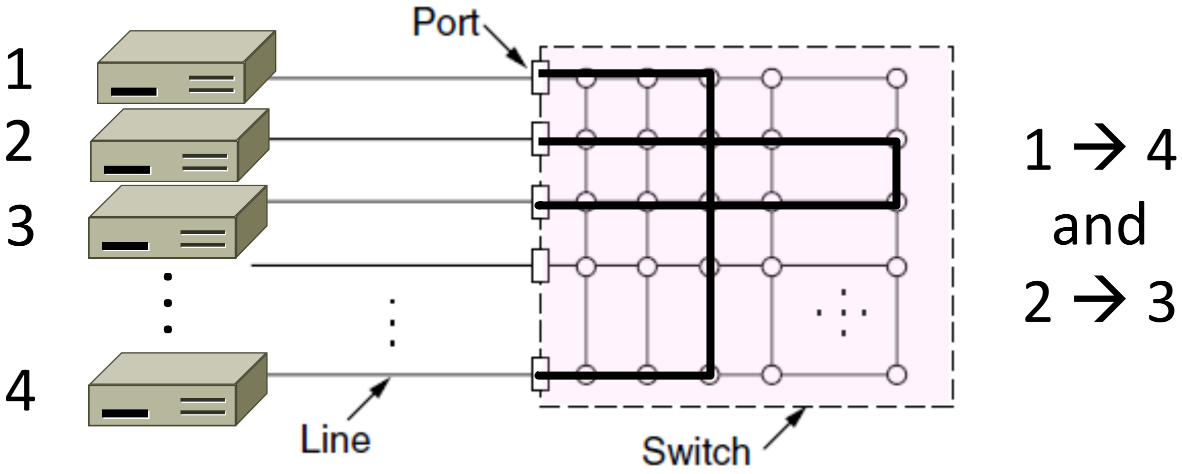
Inside a Switch

- Uses frame addresses to connect input port to the right output port; multiple frames may be switched in parallel



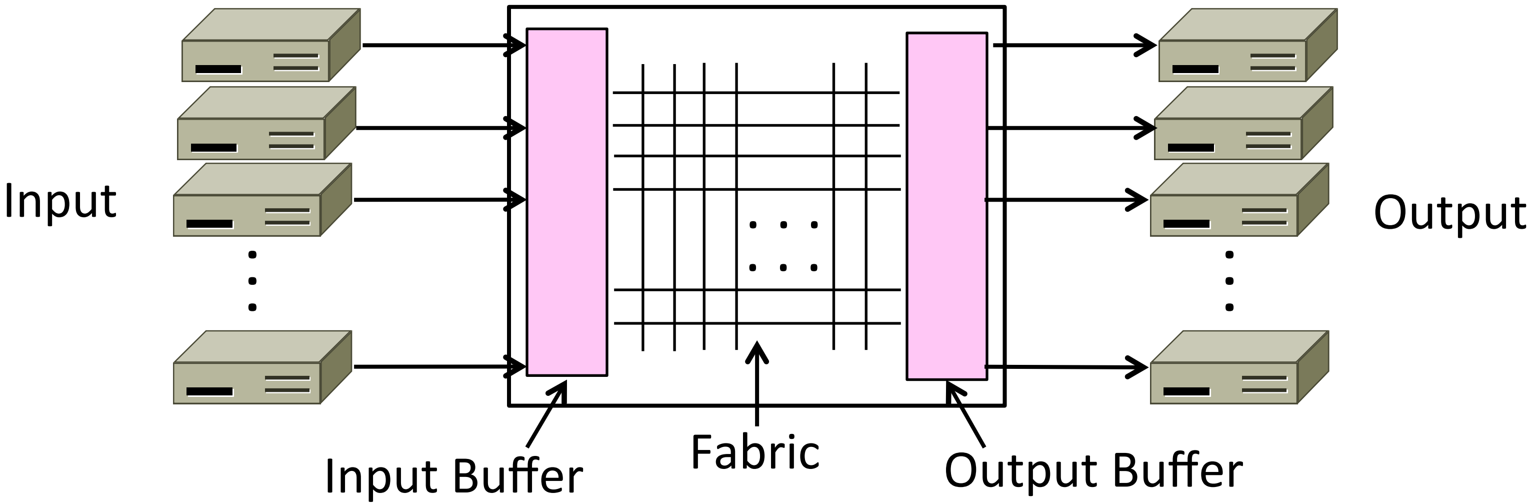
Inside a Switch (2)

- Port may be used for both input and output (full-duplex)
  - Just send, no multiple access protocol



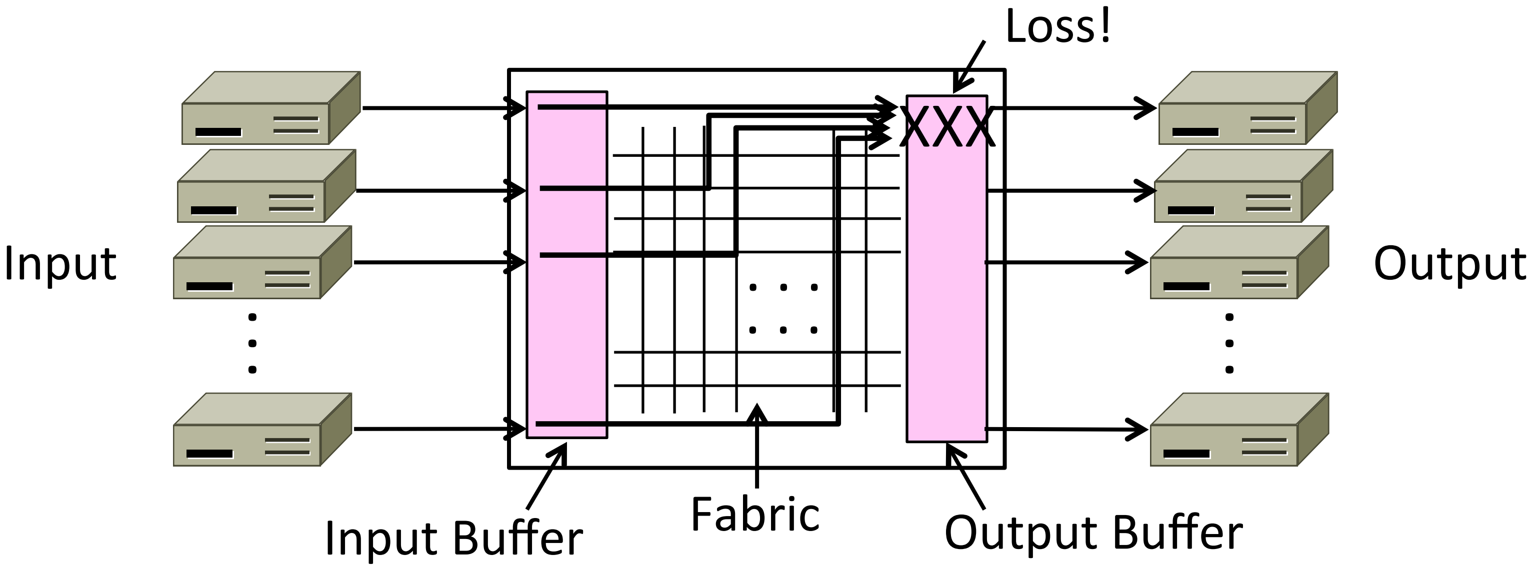
Inside a Switch (3)

- Need buffers for multiple inputs to send to one output



Inside a Switch (4)

- Sustained overload will fill buffer and lead to frame loss



Advantages of Switches

- Switches and hubs have replaced the shared cable of classic Ethernet
  - Convenient to run wires to one location
  - More reliable; wire cut is not a single point of failure that is hard to find

- Switches offer scalable performance
  - E.g., 100 Mbps per port instead of 100 Mbps for all nodes of shared cable / hub